

**IN THE CLAIMS:**

1. (Cancelled)
2. (Previously presented) A method in a data processing system for scheduling the execution of a plurality of commands, said data processing system including an environment which executes commands concurrently, wherein said commands are executed without regard to a completion of execution of any other ones of said commands, said method comprising the steps of:
  - selecting said plurality of commands from the environment which executes commands concurrently;
  - scheduling execution of said selected plurality of commands in a programming order, said scheduling step comprising:
    - encapsulating said first one of said plurality of commands in a first process and
    - encapsulating said second one of said plurality of commands in a second process;
    - beginning processing of said first process;
    - executing said first one of said plurality of commands in response to said beginning processing of said first process, wherein said first one of said plurality of commands executes only while said first process is executing; and
    - beginning processing of said second process only in response to a completion of processing of said first process.
3. (Original) The method according to claim 2, further comprising the step of completing processing of said first process in response to a completion of execution of said first one of said plurality of commands.
4. (Original) The method according to claim 2, further comprising the step of executing said second one of said plurality of commands in response to said beginning processing of said second process.
5. (Original) The method according to claim 2, further comprising the step of determining whether said first process is currently executing.

6. (Currently amended) The method according to claim 5, wherein said step of determining whether said first process is currently executing further comprises the steps of:  
establishing a return code variable for the first process; and  
utilizing said return code variable to indicate whether said first process is currently executing.
7. (Original) The method according to claim 6, wherein said step of determining whether said first process is currently executing further comprises the steps of:  
assigning a first process identifier to said first process; and  
utilizing said first process identifier to determine whether said first process is currently executing.
8. (Original) The method according to claim 7, further comprising the steps of:  
searching a process table for said first process identifier;  
determining that said first process is executing in response to locating said process identifier in said process table; and  
determining that said first process is not executing in response to a failure to locate said process identifier in said process table.
9. (Original) The method according to claim 8, further comprising the steps of:  
setting said return code variable equal to a first value while said first process is executing;  
and  
setting said return code variable equal to a second value when said first process has completed executing.
10. (Original) The method according to claim 9, further comprising the steps of:  
establishing a timer for said first process;  
starting said timer in response to executing said first process; and  
testing said return code variable to determine whether said return code variable is equal to said second value upon the expiration of said timer.

11. (Cancelled)

12. (Previously presented) A data processing system for scheduling the execution of a plurality of commands, said data processing system including an environment which executes commands concurrently, wherein said commands are executed without regard to a completion of execution of any other ones of said commands, comprising:

selecting means for selecting said plurality of commands from the environment which executes commands concurrently;

a scheduler for scheduling execution of said selected plurality of commands in said environment, said scheduler comprising:

means for encapsulating said first one of said plurality of commands in a first process and encapsulating said second one of said plurality of commands in a second process;

means for beginning processing of said first process;

means for executing said first one of said plurality of commands in response to said beginning processing of said first process, wherein said first one of said plurality of commands executes only while said first process is executing; and

means for beginning processing of said second process only in response to a completion of processing of said first process.

13. (Original) The system according to claim 12, further comprising means for completing processing of said first process in response to a completion of execution of said first one of said plurality of commands.

14. (Original) The system according to claim 12, further comprising means for executing said second one of said plurality of commands in response to said beginning processing of said second process.

15. (Original) The system according to claim 12, further comprising means for determining whether said first process is currently executing.

16. (Currently amended) The system according to claim 15, wherein said means for determining whether said first process is currently executing further comprises:  
means for establishing a return code variable for the first process; and  
means for utilizing said return code variable to indicate whether said first process is currently executing.
17. (Original) The system according to claim 16, wherein said means for determining whether said first process is currently executing further comprises:  
means for assigning a first process identifier to said first process; and  
means for utilizing said first process identifier to determine whether said first process is currently executing.
18. (Original) The system according to claim 17, further comprising:  
means for searching a process table for said first process identifier;  
means for determining that said first process is executing in response to locating said process identifier in said process table; and  
means for determining that said first process is not executing in response to a failure to locate said process identifier in said process table.
19. (Original) The system according to claim 18, further comprising:  
means for setting said return code variable equal to a first value while said first process is executing; and  
means for setting said return code variable equal to a second value when said first process has completed executing.
20. (Original) The system according to claim 19, further comprising:  
means for establishing a timer for said first process;  
means for starting said timer in response to executing said first process; and  
means for testing said return code variable to determine whether said return code variable is equal to said second value upon the expiration of said timer.

21. (Cancelled)

22. (Previously presented) A computer program product for scheduling the execution of a plurality of commands, said data processing system including an environment which executes commands concurrently, wherein said commands are executed without regard to a completion of execution of any other ones of said commands, said computer program product comprising:

selecting means for selecting said plurality of commands from the environment which executes commands concurrently;

a scheduler for scheduling execution of said selected plurality of commands in said environment, said scheduler comprising:

instruction means for encapsulating said first one of said plurality of commands in a first process and encapsulating said second one of said plurality of commands in a second process;

instruction means for beginning processing of said first process;

instruction means for executing said first one of said plurality of commands in response to said beginning processing of said first process, wherein said first one of said plurality of commands executes only while said first process is executing; and

instruction means for beginning processing of said second process only in response to a completion of processing of said first process.

23. (Original) The computer program product according to claim 22, further comprising instruction means for completing processing of said first process in response to a completion of execution of said first one of said plurality of commands.

24. (Original) The computer program product according to claim 22, further comprising instruction means for executing said second one of said plurality of commands in response to said beginning processing of said second process.

25. (Original) The computer program product according to claim 22, further comprising instruction means for determining whether said first process is currently executing.

26. (Currently amended) The computer program product according to claim 25, wherein said instruction means for determining whether said first process is currently executing further comprises:

instruction means for establishing a return code variable for the first process; and

instruction means for utilizing said return code variable to indicate whether said first process is currently executing.

27. (Original) The computer program product according to claim 26, wherein said instruction means for determining whether said first process is currently executing further comprises:

instruction means for assigning a first process identifier to said first process; and

instruction means for utilizing said first process identifier to determine whether said first process is currently executing.

28. (Original) The computer program product according to claim 27, further comprising:

instruction means for searching a process table for said first process identifier;

instruction means for determining that said first process is executing in response to locating said process identifier in said process table; and

instruction means for determining that said first process is not executing in response to a failure to locate said process identifier in said process table.

29. (Original) The computer program product according to claim 28, further comprising:

instruction means for setting said return code variable equal to a first value while said first process is executing; and

instruction means for setting said return code variable equal to a second value when said first process has completed executing.

30. (Original) The computer program product according to claim 29, further comprising:

instruction means for establishing a timer for said first process;

instruction means for starting said timer in response to executing said first process; and

instruction means for testing said return code variable to determine whether said return code variable is equal to said second value upon the expiration of said timer.

31. (Currently amended) The method according to Claim ~~1~~ 2, wherein said first process and said second process are included in a script.
32. (Previously presented) The system according to Claim 12, wherein said first process and said second process are included in a script.
33. (Previously presented) The computer program product according to Claim 22, wherein said first process and said second process are included in a script.